Writing a Fun4All Module for sPHENIX Simulations

A Case Study in Jet Reconstruction

Javier Orjuela-Koop
(Colorado)

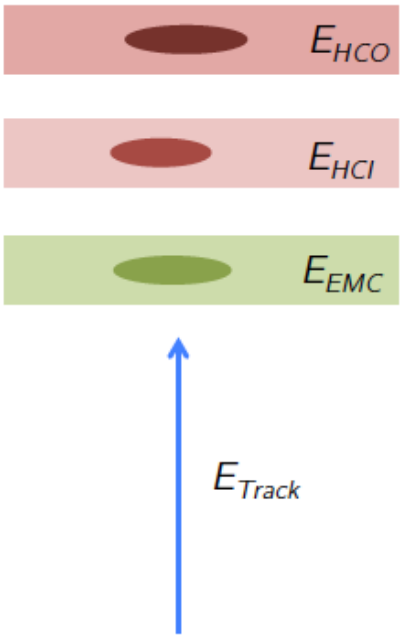
Simulations Tutorial – Part II
May Core Week 2015

Motivation

Admittedly, many people are already familiar with Fun4All.
However, here I want to bring your attention to the idiosyncrasies of sPHENIX SW and to provide a working example of the basic building blocks to get you started.

- Getting tracks and clusters from the node tree
- Iterating over tracks and clusters
- Running FastJet for jet reconstruction

Context – Particle Flow Jets

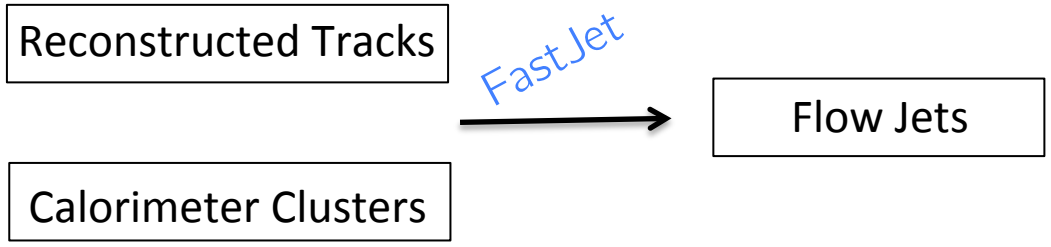


We can get improved jet energy resolution by using tracks for jet reconstruction rather than calorimeter clusters.

This requires matching individual tracks to clusters in each calorimeter layer and carefully keeping track of energy deposition.

PHFlowJetMaker

At the end of the day:
User registers the module in Fun4All_G4_sPHENIX.C and the flow jets will be on the node tree, ready for analysis



Motivation

macros/g4simulation/Fun4All_G4_sPHENIX.C

- Define a specific detector configuration (see Chris' talk)
- Read input in either HepMC format (slow) or G4Hits (fast!)
- Put clusters, tracks, towers in the Node Tree
- Evaluator output (NTuples) can be written to file.

What if you want to grab objects directly from the Node Tree and analyze them on the fly on a per-event basis?
Write a Fun4All module.

How-To

Building a library...

Makefile.am

configure.in

autogen.sh

PHFlowJetMakerLinkDef.h

PHFlowJetMaker.C

PHFlowJetMaker.h

Code in Git

`https://github.com/sPHENIX-Collaboration/tutorials/tree/master/jetreco/src`

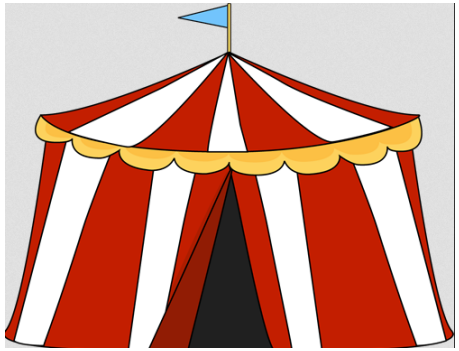
How-To

```
TOP (PHCompositeNode)/
   DCM (PHCompositeNode)/
   DST (PHCompositeNode)/
      PHG4INEVENT (PHDataNode)
      G4HIT_HCALOUT (PHIODataNode)
      G4HIT_ABSORBER_HCALOUT (PHIODataNode)
      G4HIT_SVTX (PHIODataNode)
      G4HIT_SVTXSUPPORT (PHIODataNode)
      G4HIT_CEMC (PHIODataNode)
      G4HIT_HCALIN (PHIODataNode)
      G4HIT_ABSORBER_HCALIN (PHIODataNode)
      G4HIT_HCALIN_SPT (PHIODataNode)
      G4HIT_MAGNET_0 (PHIODataNode)
      G4HIT_BH_1 (PHIODataNode)
      G4TruthInfo (PHIODataNode)
      PHHepMCGenEvent (PHIODataNode)
      G4CELL_SVTX (PHIODataNode)
      G4CELL_CEMC (PHIODataNode)
      G4CELL_HCALIN (PHIODataNode)
      G4CELL_HCALOUT (PHIODataNode)
      TOWER_CEMC (PHIODataNode)
      CLUSTER_CEMC (PHIODataNode)
      TOWER_HCALIN (PHIODataNode)
      CLUSTER_HCALIN (PHIODataNode)
      TOWER_HCALOUT (PHIODataNode)
      CLUSTER_HCALOUT (PHIODataNode)
      SVTX (PHCompositeNode)/
         SvtxHitMap (PHIODataNode)
         SvtxClusterMap (PHIODataNode)
         SvtxTrackMap (PHIODataNode)
         SvtxVertexMap (PHIODataNode)
   RUN (PHCompositeNode)/
      CYLINDERGEOM_SVTX (PHIODataNode)
      CYLINDERGEOM_SVTXSUPPORT (PHIODataNode)
      CYLINDERGEOM_HCALIN_SPT (PHIODataNode)
      CYLINDERGEOM_MAGNET_0 (PHIODataNode)
      CYLINDERGEOM_BH_1 (PHIODataNode)
      CYLINDERGEOM_CEMC (PHIODataNode)
      CYLINDERGEOM_HCALIN (PHIODataNode)
      CYLINDERGEOM_HCALOUT (PHIODataNode)
      CYLINDERCELLGEOM_SVTX (PHIODataNode)
      CYLINDERCELLGEOM_CEMC (PHIODataNode)
      CYLINDERCELLGEOM_HCALIN (PHIODataNode)
      CYLINDERCELLGEOM_HCALOUT (PHIODataNode)
      TOWERGEOM_CEMC (PHIODataNode)
      TOWERGEOM_HCALIN (PHIODataNode)
      TOWERGEOM_HCALOUT (PHIODataNode)
   PAR (PHCompositeNode)/
      SVTX (PHCompositeNode)/
         SvtxBeamSpot (PHIODataNode)
```

In Fun4All_G4_sPHENIX.C

```
gSystem->Load("libPHFlowJetMaker.so");
```

```
Fun4AllServer *se = Fun4AllServer::instance();
PHFlowJetMaker *ana = new PHFlowJetMaker(outputFile);
se->registerSubsystem(ana);
```



This macro runs the show…

How-To

Read clusters and reconstructed tracks from Node Tree

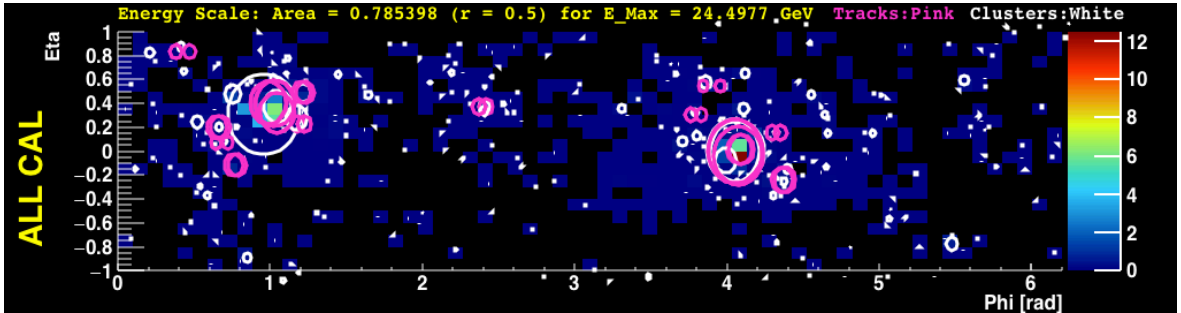
```
//Get calorimeter clusters from node tree
RawClusterContainer *emc_clusters = findNode::getClass<RawClusterContainer>(topNode,"CLUSTER_CEMC");
RawClusterContainer *hci_clusters = findNode::getClass<RawClusterContainer>(topNode,"CLUSTER_HCALIN");
RawClusterContainer *hco_clusters = findNode::getClass<RawClusterContainer>(topNode,"CLUSTER_HCALOUT");

//Get reconstructed tracks from nodetree
SvtxTrackMap* reco_tracks = findNode::getClass<SvtxTrackMap>(topNode,"SvtxTrackMap");
```

NOTE: Clustering algorithms

RawClusterBuilder → Default clustering in sPHENIX
RawClusterBuilderv1 → Sasha's clustering as used in PHENIX

`macros/g4simulations/G4_Cemc_Spacal.C`



How-To

Iterate over tracks

```
//Loop over all tracks
for(SvtxTrackMap::Iter iter = reco_tracks->begin(); iter != reco_tracks->end(); ++iter)
  {
    SvtxTrack *trk = &iter->second;
    px = trk->get3Momentum(0);
    py = trk->get3Momentum(1);
    pz = trk->get3Momentum(2);
    pt = sqrt(px*px + py*py);
    p = sqrt(px*px + py*py + pz*pz);
    track_energy = TMath::Sqrt(p*p + 0.139*0.139); //Assume pion mass
    phi = atan2(py,px);
    eta = -log(tan(acos(pz/p)/2.0));
    et = track_energy/cosh(eta);

    //Quality cut on tracks
    if(trk->getQuality() > 3.0) continue;

                ...

  }
```

How-To

Iterate over clusters

```
//Loop over EMCAL clusters
for(unsigned int i=0; i<emc_clusters->size(); i++)
  {
    RawCluster* part = emc_clusters->getCluster(i);
    double eta = part->get_eta();
    double phi = part->get_phi();
    double energy = part->get_energy()/sfEMCAL;
    double eT = energy/cosh(eta);
    double pz = eT*sinh(eta);

    if(eT<0.000001)
      {
        eT = 0.001;
        pz = eT*sinh(eta);
        energy = sqrt(eT*eT + pz*pz);
      }

    fastjet::PseudoJet pseudoJet(eT*cos(phi),eT*sin(phi),pz,energy);
    particles.push_back(pseudoJet);
  }
```

How-To

Reconstruct Jets

First define some parameters for FastJet

```
fastjet::Strategy strategy = fastjet::Best;
fJetAlgorithm = new fastjet::JetDefinition (fastjet::antikt_algorithm,r_param,strategy);
```

r_param = 0.3

vector<fastjet::PseudoJet>

```
fastjet::ClusterSequence jet_finder_raw(raw_cluster_particles,*fJetAlgorithm);
vector<fastjet::PseudoJet> raw_cluster_jets = jet_finder_raw.inclusive_jets(min_jet_pT);
```

0.5 GeV/c

How-To

```
git clone https://github.com/sPHENIX-Collaboration/tutorials/
tree/master/jetreco/src
```

From some appropriate build directory, run

```
<path to PHFlowJetMaker>/autogen.sh --prefix=<your own lib directory>
```

Remember to have your library directory appended to $LD_LIBRARY_PATH

```
setenv LD_LIBRARY_PATH {$LD_LIBRARY_PATH}:<your install lib>
```

Build and install PHFlowJetMaker

```
make install
```

```
git clone https://github.com/sPHENIX-Collaboration/macros/tree/
master/macros/g4simulations
```

Go to macros directory and run

```
root Fun4All_G4_sPHENIX_JetRecoTutorial.C
```

Questions?
Comments?